Pandoc入門：MarkdownからHTML・PDF・Writer/Word文書・スライドを生成する

OSC京都2017 1日目 セミナー

藤原 由来（日本Pandocユーザ会）

2017年8月4日

この発表について

- 文書変換ツール「Pandoc」の入門セミナーです
- 対象者
  - ドキュメンテーション（文書の作成・処理）に興味のある方
  - 文書の作成・処理を効率化したい方
  - Markdownなどの軽量マークアップ言語をうまく活用したい方
- 前提とする知識：基本的なコマンドラインの使い方
  - ターミナル (Linux/Mac)
  - コマンドプロンプト (Windows)

自己紹介

- 名前
  - 藤原 由来 (本名)
    - GitHub
    - Facebook
  - すかいゆき・藤原 惟
    - Twitter
- 職業
  - フリープログラマ・技術ライター
  - 専門学校 非常勤講師

Pandocに関する活動

- Pandocユーザーズガイドを和訳
  - Pandoc ユーザーズガイド 日本語版

    - バージョンが古くなったので、改訂を予定
- Qiita等に記事執筆
    - 多様なフォーマットに対応！ドキュメント変換ツールPandocを知ろう - Qiita
- 日本Pandocユーザ会
    - 最近Slack作りました: Slack登録フォーム

Pandoc公式サイト

- Pandoc - About pandoc
- ユーザーズガイド
    - Pandoc - Pandoc User's Guide
    - Pandoc ユーザーズガイド 日本語版

いきなりですが質問です

Q1: 普段はどんなファイルやドキュメントを扱っていますか？

- HTML
- LaTeX (数式の入ったドキュメント)
- XML系のドキュメント
- プログラミング言語のドキュメント機能
    - Javadoc, Python docstringなど
- Microsoft Office
    - Word, Excel, PowerPoint
- LibreOffice (Apache OpenOffice)

Q1: 普段はどんなファイルやドキュメントを扱っていますか？

- 軽量マークアップ言語
    - Markdown
    - reStructuredText (Sphinx)
    - Emacs org-mode
    - Wiki記法

        - MediaWiki (Wikipedia)
- その他

Q2: どのような目的でドキュメントを扱っていますか？

- Webへの公開（CMS・ブログ・Wiki含む）
- 組織内の情報共有
    - 社内Wiki
    - プロジェクト管理（ガントチャート・UMLも含む）
- 組織外との情報共有・コミュニケーション
- 自分で読み返すためのメモ
- その他

この発表でやること

- Pandocの概要
- Pandocをインストールする
- Pandocの基本的な使い方
    - Markdown ↔ LibreOffice Writer を例に
- Pandocの応用
- まとめ・お知らせ

この発表でやらないこと

- プログラミング言語のドキュメント機能
    - Pandoc自体はHaskell用ドキュメンテーションに対応
        - Haddock, Literate Haskell
    - Pandocとの組み合わせができる場合もある
- 表形式のドキュメント（Excel, CSVなど）
    - 現状のPandocが扱える文書モデルから離れるので
    - 「文書の一部」として扱うことはPandocで可能です

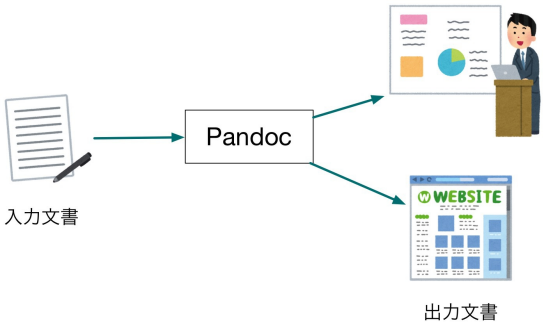
Pandocの概要

こんなことに困っていませんか？

- MS Word/LibreOffice Writer
  - 重いのでテキストファイルで下書きしたい
  - バージョン管理をしたいけど、Word文書はGit管理が面倒
- オフィスにある大量の文書を別の書式に変換したい
- MediaWiki記法で書いた原稿をSphinx(reST)で使いたい
- Markdownでスライドショーを作りたい

Pandocとは

- 文書変換ツール
  - あるフォーマットで書かれた文書を、別のフォーマットに変換するツール
  - Pandocの特徴は、対応フォーマットが非常に多いこと
- Pandoc公式サイト
  - 「a universal document converter」
  - 汎用ドキュメントコンバータ
- オープンソースソフトウェアの1つ
  - ソースコード: jgm/pandoc
  - ライセンス: GPL2

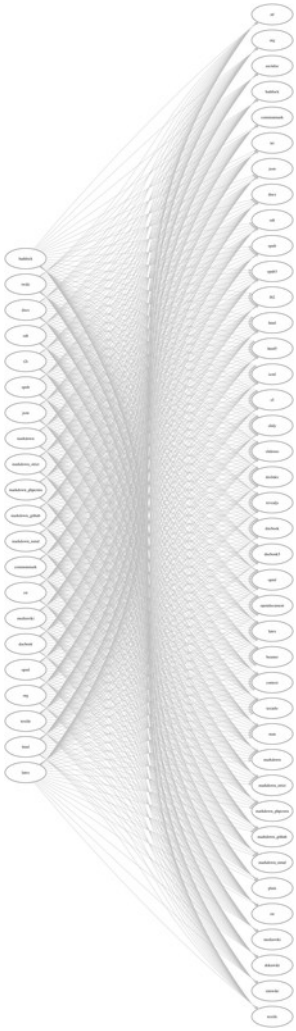


Pandocの処理フロー

対応フォーマット（一部省略）

- 入力
  - **Markdown** (Pandoc, CommonMark, PHP Markdown Extra, GitHub-Flavored Markdown, MultiMarkdown)
  - (subsets of) Textile, **reStructuredText, HTML, LaTeX, MediaWiki markup**, Emacs Org mode
  - OPML, DocBook, EPUB, **ODT** and Word docx

- 出力
  - 入力フォーマットのほとんど（ODT含む）
  - **Markdown**
  - manページ, AsciiDoc, InDesign ICML
  - **プレゼンテーション**: LaTeX Beamer, HTML5(reveal.jsなど)
  - PDF (wkhtmltopdfまたはLaTeXエンジンが必要)



Markdown？

そもそもMarkdownって何？

- このスライド自体が、実はMarkdownで書かれています
- 元々はJohn Gruberが作ったオリジナルの処理系 でHTMLに変換する

ための記法だった
- そのうちGitHubやPHPなどで記法が拡張された
  - MultiMarkdownやPandocの登場をきっかけに、目的も「論文」「プレゼンテーション」「電子書籍」など用途が広がった
  - 数々の「方言」がある状態
- 基本のMarkdownだけを覚えれば、大抵は方言が違っても「大まかには」書ける
  - おすすめ早見表: Markdown Reference (CommonMark)
  - 足りない部分は、各ツール・サービスのドキュメントを参照
  - プレビューを行うのが鉄則

裏方としてのPandoc

- 実は裏でPandocが動いているケースもいくつかあります
- R: R Markdown
  - RStudioという統合環境の中で使える
  - 厳密にはknitrの機能
- Python: Jupyter Notebook
  - nbconvert
- テキストエディタ: Typora
  - Markdownエディタの一つ (Win/Mac/Linux)
  - コマンド苦手な人でも、Pandocの変換機能を使えます

Pandocでできないこと

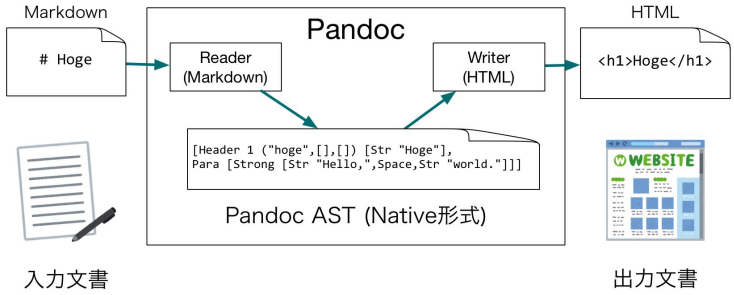
- 表主体の文書を扱うこと
  - Excel, LibreOffice Calc
  - 一部に簡単な表を埋め込むことはできる（HTMLの`<table>`相当）
- PowerPoint/LibreOffice Impressに変換すること
  - LaTeX Beamer/HTMLプレゼンには変換可能

Pandocを使う心得

- 過剰な期待をし過ぎないこと
  - Pandocは万能でないし、文書仕様の全てを満たしているわけではない
- 補助的に使うのがベスト
  - Pandocで、テキストと大まかな構造を抽出
  - 変換し切れなかった部分を、手作業や自作スクリプトで編集

Pandocの実装

- 言語: Haskell
  - Pandoc的には、「厳密に型が定義されている」ことがありがたい
  - Haskellは構文解析器(パーサ)を作るのにすごく適している(Parsecなど)
- モジュール構成
  - Reader: 入力文書を解析し、Haskell上の中間文書に変換する
  - Writer: 中間文書を受け取り、出力フォーマットに変換する



Pandocの処理フロー（詳細）

PandocにおけるMarkdown

Pandocが扱えるMarkdown方言

- Pandoc's Markdown: `-f markdown`
  - Pandocにおける標準のMarkdown方言
  - 技術文書から論文・電子書籍まで幅広く対応
- GitHub Flavored Markdown (gfm): `-f markdown_github`
  - GitHubの標準、プログラマ・フレンドリーな方言
- PHP Markdown Extra: `-f markdown_phpextra`
  - 最近はMarkdown Extraとも呼ばれる
- MultiMarkdown: `-f markdown_mmd`
  - HTMLだけでなくLaTeXなどの論文も意図した処理系
- CommonMark: `-f commonmark`
  - 仕様の曖昧さをなくすことを目的とした仕様/処理系
  - 事実上の標準？（RFCなどによる正式な標準ではない）

Pandoc's Markdownの特徴

- 詳しくは Pandoc's Markdown を参照
  - または Pandoc ユーザーズガイド 日本語版 を参照
- HTMLの定義リスト(`<dl>`, `<dt>`, `<dd>`)がある
- 表(`<table>`)は4種類ある
  - 日本語には「Grid Table」か「Pipe Table」がおすすめ
- **ヘッダ部分にメタデータを記述できる（重要）**
  - タイトルブロック（行を`%`で始める）
    - タイトル（1行目）、著者（2行目）、日付（3行目）のみ簡潔に書ける
  - YAMLメタデータブロック（次のスライド）

YAMLメタデータブロック

- ブロックを`---`で始めて`...`で閉じる
- 改行などの書き方は、YAMLの文法に従う
  - 参考: Rubyist Magazine - プログラマーのための YAML 入門 (初級編)
- このブロックで定義されたデータは、メタデータという種類の変数となる
  - メタデータは文書変換する際のオプションや制御に使うことができる
    - `pandoc -D (出力書式の名前)`で、実際の使われ方がわかる
  - このブロックを使えば、文書自体に文書変換のオプションを埋めこめる（便利）

YAMLメタデータブロックの例

```
---
title: Pandoc入門：MarkdownからHTML・PDF・Writer/Word文書・スライドを生成する
author: 藤原 由来
date: 2017年8月4日
revealjs-url: reveal.js-3.4.0
theme: sky-sky-y
transition: fade
transitionSpeed: fast
slideNumber: true
...
```

Pandocをインストールする

ターミナルを開く

- Linux/Mac: ターミナル
- Windows: コマンドプロンプト
  - 分かっている方は、お好きなターミナル・シェルでもOK
- 基本的なコマンド操作については、今回は説明しません
  - コマンドが苦手な方は「何ができるか」を覚えてもらえれば幸いです

Pandocのインストール: インストーラ編

- Windows/Macの場合
- パッケージを直接落としてインストール
  1. ここからパッケージをダウンロード
     - Windows: `.msi`, Mac: `.pkg`
  2. インストール

Pandocのインストール: パッケージマネージャ編

- Mac(Homebrew)
  - `$ brew install pandoc`
- Windows(Chocolatey)
  - `> cinst -y pandoc`

Pandocのインストール: Linux編

- Linux
  - pandoc/INSTALL.md を参照
  - 各種パッケージマネージャでインストールできます
    - Debian, Ubuntu, Slackware, Arch, Fedora, NiXOS, openSUSE, and gentoo
    - 各々のパッケージマネージャで「pandoc」をsearch/installしてください
    - バージョンが古いことがあるので注意

    - ソースコードからビルド
        - Haskellのソースコードをビルドする必要があります
        - Stack(Haskellビルドツール)を
        - ソースコード: GitHub - jgm/pandoc

---

wkhtmltopdfのインストール

- PDF出力のために必要
    - TeXLiveを入れていれば、LaTeX処理系も利用可能（説明略）
    - ただし、pLaTeXはNGなので、LuaLaTeX/XeLaTeXが必要です
- インストーラを直接落としてインストール
    1. wkhtmltopdf - Downloadsからダウンロード
    2. インストール
- パッケージマネージャでインストール
    - Mac(Homebrew): `$ brew cask install wkhtmltopdf`
        - Caskの方なので注意
    - Windows(Chocolatey): `> cinst -y wkhtmltopdf`
    - Linux: 略（各々のパッケージマネージャで「wkhtmltopdf」をsearch/installしてください）

動作確認1: Pandoc単体

※ 藤原の環境：Windows (Chocolatey) + MSYS2

```
$ pandoc --version
$ pandoc --list-input-formats
$ pandoc --list-output-formats
$ echo "**Hello**" | pandoc -f markdown -t html
<p><strong>Hello</strong></p>
```

動作確認1: Pandoc単体

```
$ echo "**Hello**" | pandoc -f markdown -t html
```

- シェルの**パイプ**機能を使っています
    - `echo`が出す標準出力をパイプ (|)で`pandoc`の標準入力に渡す
    - `pandoc`は入力・出力ファイル名が与えられてない場合、標準入力・標準出力を使う
- `-f`: 入力フォーマット名 (from)
    - 使えるフォーマット名は `pandoc --list-input-formats` で確認できる
- `-t`: 出力フォーマット名 (to)
    - 使えるフォーマット名は `pandoc --list-output-formats` で確

認できる

動作確認2: ファイルを入力

- 次の内容をテキストファイルで保存し、「hello.md」と保存する

```
# Hello
こんにちは
```

動作確認2: ファイルを入力

- コマンドを実行:

```
$ pandoc hello.md -o hello.html
```

- オプションのない引数(hello.md): 入力ファイル名
- `-o`: 出力ファイル名 (output)
  - `-t`（次スライド）を指定しない場合、拡張子から出力フォーマットを推測してくれる
- **注意: Pandocが対応している文字コードはUTF-8のみです**
  - UTF-8以外を扱う場合は、nkf/iconv/uconvなどの文字コード変換ツールをパイプ(|)に繋げます

動作確認3: Pandoc + wkhtmltopdf (PDF)

```
$ echo "**Hello**" | pandoc -f markdown -t html5 -o hello.pdf
```

- `-f`: 入力フォーマット名 (from)
- `-t`: 出力フォーマット名 (to)
  - 注意: wkhtmltopdfでPDFを出すときは `-t html5`を指定
    - 内部で文字通り、HTML5に変換してからPDFに出すので
- `-o`: 出力ファイル名 (output)
  - 注意: wkhtmltopdfでPDFを出すときは、`-o`の拡張子は`.pdf`を指定

Pandocの基本的な使い方

これからやること

- Markdown ↔ LibreOffice Writer の相互変換を例にします
  - 他の書式に変換するときの基礎になります
  - MS Wordを扱うときは、ほぼ同じです

これからやること

- Markdown文書からWriter文書に変換する
  - とりあえず変換してみる
  - 綺麗なWriter文書を生成する: reference機能
- Writer文書をMarkdownなどに変換する
- 以下の作業では、GitHubリポジトリのsampleディレクトリにあるファイルを使います
- atarashii_kenpo.md: あたらしい憲法のはなし(Markdown版)より
  - nogajunさん編、Public Domain

とりあえず変換してみる: pandocコマンド

```
$ git clone https://github.com/sky-y/osc-kyoto2017-pandoc.git
$ cd sample
$ pandoc atarashii_kempo.md -o atarashii_kempo.odt
```

- -o: 出力ファイル名
  - 多くの場合拡張子で判断できる

ファイルを開く

```
$ open atarashii_kempo.odt      # Mac
$ xdg-open atarashii_kempo.odt  # Linux (GNOME, KDE, Xfce)
> start atarashii_kempo.odt     # Windows
```

綺麗なWriter文書を生成する

- Pandocのreference機能を使う
  1. Pandocコマンドからreferenceファイルを生成
  2. reference.odtをWriterで編集する

    - 3. Pandocの変換時にreference.odtをオプションで指定する
        - もしくはユーザデータディレクトリに入れる
- Wordの場合は、「odt」を「docx」に読み替えると同様にできます

---

(1) Pandocコマンドからreferenceファイルを生成

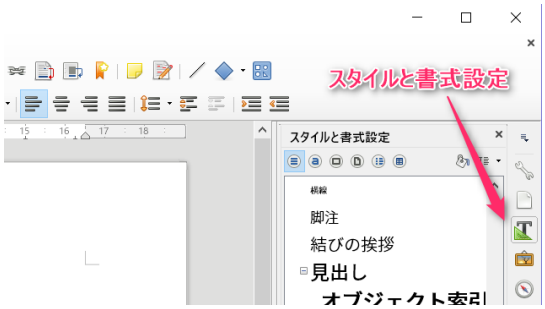
```
$ pandoc --print-default-data-file reference.odt > reference.odt
```

- コマンドオプションで指定する場合は「reference.odt」という名前でなくてよい
- ユーザデータディレクトリ(後述)に入れる場合は必ず「reference.odt」という名前にする

---

(2) reference.odtをWriterで編集する

- reference.odtをLibreOfficeで開く
- reference.odtの内容はPandocから参照されない
    - デフォルトで「Hello World!」と表示されている部分のこと
    - 例えば表示用サンプルを置いてもいい
- 「スタイルと書式設定」から書式を変更



---

(3) Pandocの変換時にテンプレートをオプションで指定する

注意: バージョンによって使用するオプションが違います

- `--reference-odt`: Pandoc 1.xの指定
- `--reference-doc`: Pandoc 2.xの指定
- バージョンは `pandoc -v` で分かります

- 実際に使えるコマンドは `pandoc -h`で分かります
  - UNIX系なら `pandoc -h | grep 'reference'` で絞れるはず

```
$ pandoc atarashii_kenpo.md --reference-odt=reference.odt -o
atarashii_kenpo.odt
$ pandoc atarashii_kenpo.md --reference-doc=reference.odt -o
atarashii_kenpo.odt
```

補足: reference.odtをユーザデータディレクトリに入れる

- ユーザデータディレクトリの場所: `pandoc -v` で出る
  - Windows: `C:\Users\ユーザ名\AppData\Roaming\pandoc`
  - Mac: `$HOME/.pandoc`
- このディレクトリに「reference.odt」という名前でテンプレートを入れると、次回からデフォルトで使ってくれる

具体的なノウハウ

- nogajunさん: PandocとLibreOffice WriterでiDエディタのマニュアルを製本する, どうしてこうなった - Days of Speed(2014-12-06)
  - nogajun/pandoc-writerの`pandoc-writer.odt`がテンプレートとして使える
- いくやさん: PandocでMarkdownをODTに変換する - いくやの斬鉄日記
  - 画像のサイズを整える (ImageMagickの`mogrify`コマンド)
  - 画像のDPIを変更する (同上)

Writer文書からMarkdown/reST文書に変換してみる

- nogajunさんの`pandoc-writer.odt`を変換してみる
  - nogajun/pandoc-writer (CC BY-SA 4.0)
- Markdown (Pandoc's)
  - `$ pandoc pandoc-writer.odt -o pandoc-writer.md`
- reStructuredText (Sphinxなどで使用)
  - `$ pandoc pandoc-writer.odt -o pandoc-writer.rst`

Writer文書からLaTeX文書に変換してみる

- LaTeX (ヘッダ・フッタ抜き)
  - `$ pandoc pandoc-writer.odt -o pandoc-writer.tex`
  - あとで別のLaTeXソースファイルに貼り付けたり、スクリプトで組み入れたりするのに便利
  - PandocはpLaTeXに対応しないので、pLaTeX前提ならこちらがおすすめ
- LaTeX (ヘッダ・フッタ入り)
  - `$ pandoc -s pandoc-writer.odt -o pandoc-writer.tex`
    - `-s`: 文書として完全になるようにヘッダ・フッタを付ける (standaloneモード)
  - 注意：LuaLaTeX/XeLaTeXのみ対応
    - `$ pandoc -s pandoc-writer.odt --latex-engine=lualatex -o pandoc-writer.tex`
    - `--latex-engine=`オプションで`lualatex` or `xelatex`を指定
  - 日本語設定
    - LuaLaTeX: LuaTeX-jaの使い方
    - XeLaTeXを使う場合: BXjscls がまた新しくなった（v1.1a） - マクロツイーター

Q&A

Pandocの応用

Pandocの応用

- オフィスにある大量の文書を別の書式に変換したい
- Markdownでスライドショーを作りたい
- フィルタ機能
- おまけ：電子書籍について

オフィスにある大量の文書を別の書式に変換したい

処理したいファイルが大量にある場合は、スクリプトにPandocを組み込みます。

1. 下準備: 他のツールなどで、なんとかしてPandocが処理できる書式に変換する
    - おすすめ: HTML（多くのツールでエクスポートできるので）
2. Pandocをスクリプトの中で使う
    - シェルスクリプトで直接使う
    - スクリプト言語の外部コマンド機能で呼ぶ
    - スクリプト言語のライブラリから呼ぶ（古い場合があるので注意）

Pandocをスクリプトとして呼ぶ例（その他）

- 各種エディタのプラグイン・拡張で対応
    - Atom, VS Code, …
- 特にVimの場合
    - LaTeX文書を書くときの補助として「Markdownを書いてその場でLaTeXに変換する」拡張がある
    - TeXで書くのめんどくさい部分はmarkdownで書いちゃえば最強じゃないかな?【Vim + pandoc】 - Qiita

---

Markdownでスライドショーを作りたい

このスライド自体をPandocで生成しました

- Pandoc's Markdownの書式に従って原稿を書く
    - もちろんこれ以外の書式でも、Pandocが対応していれば書けます
- Pandocで変換する
    - 今回は「reveal.js」形式（HTML+JavaScriptによるプレゼン）に変換
    - `$ pandoc input.p.md -s -f markdown -t revealjs -o index.html`
        - 実際のファイルは次スライドで
        - `-s`: standalone (ヘッダ・フッタの付いた完全な文書を出力)
- アップロードする
    - GitHub Pagesを使うと、直接GitHubにpushすればアップロード

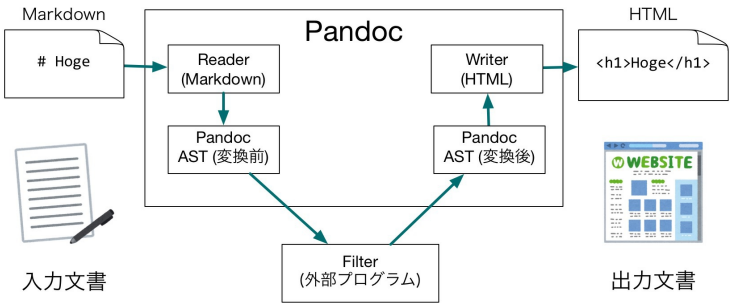
できます
        - この場合は、`.nojekyll`という空ファイルを置かないと、404エラーになるので注意
- その他 LaTeX Beamer にも変換できます

実際のソースコード

- このスライド自体がGitHub Pagesにアップロードされています
- GitHubリポジトリ
    - 発表用スライド (HTML/reveal.js)
    - スライドのMarkdownソース (Pandoc's Markdown)
    - 復習用資料 (GitHub Flavored Markdown)
    - 変換の補助にGulpを使っています（`make`のようなもの）

フィルタ機能

- 中間文書をJSON形式に出す
- それを外部スクリプトが標準入力で受け取り処理する
- それを標準出力に出して、Pandocに戻す



Pandocの処理フロー（フィルタ付き）

フィルタ機能ができること

- 詳しくはPandoc Filters (GitHub Wiki)を参照
- 引用を入れる (pandoc-citeproc, pandoc-crossref)
    - MarkdownとPandocを使って論文っぽい文章を書く | inoblog
- 図表を入れる
    - (mermaid-filter): mermaidで使えるフローチャート、ガントチャート、シーケンス図
    - pantable: CSVファイルを表として読み込む
- 外部ファイルの読み込み

    - Node.jsで書くチュートリアル: pandocでMarkdownを拡張しコードをインポート出来るfilterを書く | Web Scratch
- 過去のチュートリアル「HaskellでPandocフィルタを実装する」
    - Haskell with Skype Pandocチュートリアル 第2回

おまけ：電子書籍について

- PandocもEPUB出力できる
    - 素朴なEPUBなら日本語でも`-t epub3`で出力できる
- Markdown→EPUB変換には「でんでんコンバーター」をおすすめします
    - ルビや縦中横が使えて、細かい設定や組版がしやすい
    - 記法: でんでんマークダウン
        - PHP Markdown Extraベース
- 提案
    - Pandocに対応する好きな記法で原稿を書く
    - `pandoc -t markdown_phpextra`で、でんでんマークダウン向けに変換
    - でんでんエディターにペーストして仕上げる

まとめ・お知らせ

今日やったこと

- Pandocの概要
- Pandocをインストールする
- Pandocの基本的な使い方
    - Markdown ↔ LibreOffice Writer を例に
- Pandocの応用

Pandocの今後の課題

- 日本語に特化した文書フォーマットにほとんど対応していない
    - 書籍におけるルビや圏点など
    - 日本語コミュニティの必要性
- 表形式の文書は対応していない
    - Excel文書など→Excel方眼紙への対策には致命的
    - サードパーティのプリプロセッサにより部分的に変換する手

段はある
        - 一部の図表（Graphvizなど）はこの方法で取り込むことができる
    - 参考: Excel方眼紙公開討論会 (9/30@東京)

日本Pandocユーザ会

- Webサイト (リニューアル予定)
    - http://sky-y.github.io/site-pandoc-jp/
- Slackを始めました（**どなたでも歓迎します！**）
    - Slack登録フォーム

ドキュメンテーションWiki

- ドキュメンテーションWiki (GitHub)
- 誰でも編集歓迎します（要GitHubアカウント）

Q&A

- 連絡先
    - メールフォーム: https://goo.gl/forms/FPB22jv9S5NP4fpx2
    - Twitter: すかいゆき・藤原 惟 @sky_y
    - Facebook: https://fb.me/sky.yuki.f